# 登录系统

# 创建目录

mkdir download

mkdir local

mkdir projects

# Mysql

**Download and Install**

wget <https://cdn.mysql.com//Downloads/MySQL-5.7/mysql-5.7.25-linux-glibc2.12-x86_64.tar.gz>

tar -xvzf mysql-5.7.25-linux-glibc2.12-x86\_64.tar.gz

mv mysql-5.7.25-linux-glibc2.12-x86\_64 /data/local/mysql

**添加用户及用户分组、目录权限**

groupadd mysql

useradd mysql -g mysql

cd /data/local/mysql

chown -R mysql .

chgrp -R mysql .

**创建数据库目录**

mkdir data

chown -R mysql:mysql data

vi /etc/my.cnf

|  |
| --- |
| [mysql]  default-character-set=utf8  [mysqld] skip-name-resolve  port = 3306  basedir=/data/local/mysql  datadir=/data/local/mysql/data  max\_connections=200  character-set-server=utf8  default-storage-engine=INNODB  lower\_case\_table\_names=1 max\_allowed\_packet=16M |

yum install libaio

bin/mysql\_install\_db --user=mysql --basedir=/data/local/mysql/ --datadir=/data/local/mysql/data/

cp ./support-files/mysql.server /etc/init.d/mysqld

chmod a+x /etc/init.d/mysqld

chkconfig mysqld on

#编辑  vi /etc/profile

|  |
| --- |
| export PATH=$PATH:/data/local/mysql/bin |

source /etc/profile

#获得初始密码

cat /root/.mysql\_secret

mysql -uroot –p

set password for root@localhost=password('gElXe53blqj2');

# Java

wget https://download.oracle.com/otn-pub/java/jdk/8u201-b09/42970487e3af4f5aa5bca3f542482c60/jdk-8u201-linux-x64.tar.gz?AuthParam=1548480950\_1084dbc34a3500e4e492321407d55179

mv jdk-8u201-linux-x64.tar.gz?AuthParam=1548436859\_90497690909136a966ca815486514149 jdk-8u201-linux-x64.tar.gz

tar zxvf jdk-8u201-linux-x64.tar.gz

mkdir -p /data/local/java

mv jdk1.8.0\_201 /data/local/java

#编辑  vi /etc/profile

|  |
| --- |
| export JAVA\_HOME=/data/local/java/ jdk1.8.0\_201  export JRE\_HOME=${JAVA\_HOME}/jre  export CLASSPATH=.:${JAVA\_HOME}/lib:${JRE\_HOME}/lib  export PATH=$PATH:/data/local/mysql/bin:$JAVA\_HOME/bin |

source /etc/profile

mkdir -p /usr/java/packages/lib

cp -rf amd64 /usr/java/packages/lib

# Httpd

yum remove apr apr-util httpd

## 安装apr

wget <http://mirror.bit.edu.cn/apache//apr/apr-1.6.5.tar.gz>

tar zxvf apr-1.6.5.tar.gz

cd apr-1.6.5

./configure --prefix=/data/local/apr

make && make install

mkdir -p /home/local/apr/lib/

cp –rf /data/local/apr/lib/libapr-1.la /home/local/apr/lib/libapr-1.la

**安装apr-util**

yum install expat-devel

wget <http://mirror.bit.edu.cn/apache//apr/apr-util-1.6.1.tar.gz>

tar zxvf apr-util-1.6.1.tar.gz

cd apr-util-1.6.1

./configure --prefix=/data/local/apr-util –with-apr=/data/local/apr

make && make install

**安装pcre**

wget <ftp://ftp.csx.cam.ac.uk/pub/software/programming/pcre/pcre-8.42.zip>

yum install unzip

yum install -y gcc gcc-c++

unzip pcre-8.42.zip

cd pcre-8.42

./configure --prefix=/data/local/pcre --enable-utf8

make && make install

**安装ssl**

wget <https://www.openssl.org/source/openssl-1.0.2q.tar.gz>

tar zxvf openssl-1.0.2q.tar.gz

cd openssl-1.0.2q

./config --prefix=/data/local/openssl –fPIC

我们都知道在生成一个动态库时需要指定-fPIC，这是创建动态库所要求的，共享库被加载是在内存中的位置是不固定的，是一个相对的位置。

那么在生成静态库时通常不指定-fPIC, 可是在64bit编译使用静态库就会提示需要-fPIC重新编译该库。

由于openssl编译静态库时，没有使用-fPIC选项，使得编译出来的静态库没有重定位能力。  
这样在64bit机器上编译出来的静态库如果不指定-fPIC选项几乎全部不能使用。

因此需要重新加上-fPIC从新编译openssl

make && make install

**安装httpd**

wget <https://mirrors.tuna.tsinghua.edu.cn/apache//httpd/httpd-2.4.38.tar.gz>

tar zxvf httpd-2.4.38.tar.gz

cd httpd-2.4.38

./configure --prefix=/data/local/httpd \

--with-apr=/data/local/apr \

--with-apr-util=/data/local/apr-util \

--with-pcre=/data/local/pcre \

--enable-ssl --with-ssl=/data/local/openssl \

--enable-mods-shared=most --enable-ssl \

--enable-module=so --enable-rewrite --enable-proxy \

--enable-proxy-ajp --enable-module=most \

--enable-module=deflate \

--enable-dav-svn --enable-authz-svn \

--enable-cache --enable-disk-cache --enable-mem-cache

make && make install

## 加入开机自启动

cp /data/local/httpd/bin/apachectl /etc/init.d/httpd

vi /etc/init.d/httpd

在#!/bin/sh后面加入下面两行  
#chkconfig:345 85 15  
#description: Start and stops the Apache HTTP Server.

chmod a+x /etc/init.d/httpd

chkconfig --add httpd

chkconfig --list httpd

开启80端口

systemctl stop firewalld.service

systemctl start firewalld.service

systemctl status firewalld.service

sudo firewall-cmd --list-all

sudo firewall-cmd --add-service=http --permanent

sudo firewall-cmd --add-port=80/tcp --permanent

sudo firewall-cmd --add-port=8080/tcp --permanent

sudo firewall-cmd --add-port=8081/tcp --permanent

sudo firewall-cmd --reload

sudo firewall-cmd --list-all

# Httpd集成Tomcat

wget http://archive.apache.org/dist/jakarta/tomcat-connectors/jk/source/jk-1.2.6/jakarta-tomcat-connectors-jk-1.2.6-src.tar.gz

tar zxvf tomcat-connectors-1.2.42-src.tar.gz

cd tomcat-connectors-1.2.42-src/native

./configure --with-apxs=/data/local/httpd/bin/apxs

make

make install

# Tomcat

wget http://mirrors.shu.edu.cn/apache/tomcat/tomcat-8/v8.5.37/bin/apache-tomcat-8.5.37.tar.gz

mkdir /data/local/tomcats

tar zxvf apache-tomcat-8.5.37.tar.gz

rm –rf /data/local/tomcats/\*

cp -rf apache-tomcat-8.5.37 /data/local/tomcats/tomcat-minmeng-manager

cp -rf apache-tomcat-8.5.37 /data/local/tomcats/tomcat-minmeng-web-1

cp -rf apache-tomcat-8.5.37 /data/local/tomcats/tomcat-minmeng-web-2

cp -rf apache-tomcat-8.5.37 /data/local/tomcats/tomcat-minmeng-web-3

rm –rf /data/local/tomcats/tomcat-minmeng-manager

cp -rf apache-tomcat-8.5.37/data/local/tomcats/tomcat-minmeng-manager

**tomcat-native**

yum install tomcat-native

wget http://archive.apache.org/dist/tomcat/tomcat-connectors/native/1.2.18/source/tomcat-native-1.2.18-src.tar.gz

tar zxvf tomcat-native-1.2.18-src.tar.gz

cd tomcat-native-1.2.18-src/native/

./configure –prefix=/data/local/tomcat-native --with-apr=/data/local/apr/bin/apr-1-config --with-java-home=$JAVA\_HOME

rm -rf /data/local/tomcat-native

make && make install

vi /etc/profile

|  |
| --- |
| export LD\_LIBRARY\_PATH=$LD\_LIBRARY\_PATH: /data/local/tomcat-native/lib |

**TOMCAT启动很慢的解决办法**

1）在Tomcat环境中解决  
  
可以通过配置JRE使用非阻塞的Entropy Source。  
  
在catalina.sh中加入这么一行：-Djava.security.egd=file:/dev/./urandom 即可。  
  
加入后再启动Tomcat，整个启动耗时下降到Server startup in 2912 ms。

2）在JVM环境中解决  
  
打开$JAVA\_PATH/jre/lib/security/java.security这个文件，找到下面的内容：

securerandom.source=file:/dev/urandom

替换成

**securerandom.source=file:/dev/./urandom**